

FUNCTION WEIGHTED METRIC DISCOVERY FOR UNRELIABLE FUNCTIONS

ALEXANDER CLONINGER

ABSTRACT. We consider building a function adapted diffusion operator high dimensional data X when the function F can only be evaluated on large subsets of the data, and possibly only depends on a small subset of the features. Our method breaks X up into hierarchical trees, and determines the importance of each feature in each subset of the population. The resulting metric ρ_F is then used to define a localized filtration of F and estimation values of F at a finer scale than it is reliable naively. We apply this method in several cases in which F is unreliable a priori, and specifically use it to determine personalized risk and treatment effectiveness in drug trials. We validate the model on several synthetic datasets.

1. INTRODUCTION

We address the problem of building a metric on a high dimensional data set $X = \{x_i\} \subset \mathbb{R}^d$ that is smooth with respect to an external nonlinear function F . The types of functions $F : X \rightarrow \mathbb{R}$ we consider are functions that can only be evaluated on large subsets of the data. In other words, $F(E)$ for $E \subset X$ is only reliable (or even computable) when $|E| \geq c > 0$. We build an algorithm which constructs a metric $\rho : X \times X \rightarrow \mathbb{R}^+$ such that

$$|F(E) - F(E')| < C\rho(E, E'),$$

for a small constant C . In other words, the metric ρ does not only consider the geometry of the space X , but also the geometry and the properties of the function F being studied. We discuss examples and applications of such F in Section 1.1.

The purpose of computing ρ is two-fold:

- (1) this discovers the intrinsic organization of X which dictates changes in F . This makes any subsequent clustering or analysis done using ρ reflect the level sets of F , as well as the intrinsic structure of X . The reason for doing this is that F may not be smooth with respect to the intrinsic geometry of the space, but has structure that is well described by a subset of the features. Also,
- (2) this allows for simple estimation of F at a finer scale than it is reliable naively. Using ρ , we are able to construct an estimate of F , which we call \hat{f} , which can be

evaluated pointwise via

$$\hat{f}(x) = \lim_{\epsilon \rightarrow 0} F(\mathcal{N}_d^\epsilon(x)), \quad \mathcal{N}_\rho^\epsilon(x) = \{z \in \mathbb{R}^m : \rho(x, z) < \epsilon\}. \quad (1)$$

One can also define a multi-scale decomposition of F via

$$\hat{f}(x) = \sum_{\epsilon_i} f_{\epsilon_i}(x), \quad f_\epsilon(x) = F(\mathcal{N}_d^{\epsilon/2}(x)) - F(\mathcal{N}_d^\epsilon(x)). \quad (2)$$

The key in both these approximations is that, provided F is smooth with respect to ρ , the approximating ϵ -neighborhood will have a large radius about level sets of F . This increases the number of points $\{x_i\}$ in $\mathcal{N}_\rho^\epsilon(x)$ for a fixed ϵ , making the approximations more accurate than those generated by taking an isotropic ball of radius ϵ about x .

1.1. Use Cases of F . There are several medical applications in which F may not be computable or interpretable pointwise, but instead only exist on large subsets of patients.

- (1) **Modeling Semi-Supervised Outcomes:** Let $X = \{x_i\}$ represent the m pieces of baseline information of n people, and let $\xi : X \rightarrow \{0, 1\}$ be a random binary variable such that

$$\xi(x) = \begin{cases} 1, & \text{outcome of } x \text{ is observed} \\ 0, & \text{did not observe outcome of } x \end{cases}.$$

We only observe the outcome

$$f(x) = \begin{cases} Z_X, & \xi(x) = 1 \\ NaN, & \xi(x) = 0 \end{cases},$$

where Z_X is the outcome of patient x , and NaN implies the outcome is unknown.

This function is difficult to learn, especially since $\#\{x : \xi(x) = 1\} < n$. One can instead consider the function

$$F(x) = \mathbb{E}_{\{x \in \mathcal{N}_\rho^\epsilon(x) | \xi(x)=1\}} [f(x)], \quad \mathcal{N}_\rho^\epsilon(x) = \{z \in \mathbb{R}^m : \rho(x, z) < \epsilon\} \quad (3)$$

as the probability x has an outcome by considering their neighbors observed. Determining a metric ρ learns the level sets of F , and thus maximizes the range of F , is akin to finding pockets of people, based only on baseline information X , that are at high risk (or low risk) of an outcome. Discovery of the metric $\rho(x, y)$ then allows for a much more robust analysis of “types” of people at various risk values, as there may be large level sets of F that stretch across a diverse group of patients.

- (2) **Modeling Treatment Effectiveness:** Let $X = \{x_i\}$ represent the m pieces of baseline information of n people, and now consider that each patient has a choice of drug in $\alpha = \{A, B\}$. The treatment group is denoted $X_\alpha = 1$ if x is in treatment B , 0 otherwise. The current method of dealing with treatment effectiveness is the *Cox*

proportional hazard model [4]. In it, we let $\lambda_0(t)$ be the common baseline hazard function which describes the risk of an outcome at each time step independent of treatment. Within treatment groups, the hazard function for the Cox proportional hazard model takes the form

$$\lambda(t) = \lambda_0(t)e^{\beta X_\alpha + \tilde{\gamma} X},$$

where $\alpha \in \{A, B\}$ and X_α is an indicator function for which treatment patient x was in.

A patient personalized version of this model would be

$$\lambda(t|x) = \lambda_0(t|x)e^{g(X) + X_\alpha h(X)},$$

which now allows the benefit or detriment of the drug to be patient specific.

In this model of personalized risk, g and h are unknowable pointwise in a drug trial since each x only takes one of the drugs in α . So we estimate g and h in a neighborhood by assuming

$$\begin{aligned} \lambda_0(t|z) &\approx \lambda_0(t|x), \quad g(z) \approx g(x), \quad h(z) \approx h(x), \\ \text{for } z &\in \mathcal{N}(x) = \{z \in \mathbb{R}^m : d(x, z) < \epsilon\}. \end{aligned}$$

Thus we can run a cox proportional hazard model on $z \in \mathcal{N}(x)$ by fitting β to

$$\begin{aligned} \lambda(t|z) &= \lambda_0(t)e^{\beta Z_\alpha}, \quad z \in \mathcal{N}(x), \\ F(x) &= \beta. \end{aligned}$$

Thus, $F(x)$ reflects the amount a patient is positively or negatively affected by a drug, and can be used to approximate $h(x)$. Determining a metric ρ learns the level sets of F , and thus maximizes the range of F , is akin to finding pockets of people, based only on baseline information X , that are at much higher risk (or lower risk) on drug A than they are on drug B . Discovery of the metric $\rho(x, y)$ then allows for an analysis of “types” of responders and non-responders.

1.2. Main Contributions. The study of individualized treatment effects has recently been considered with linear lasso models [11], and linear logistic models with AdaBoost [7]. A number of models have been built to predict outcomes from a single treatment, but high risk for an outcome does not necessarily imply treatment benefit [6, 5]. While these models provide useful treatment recommendations, they project to a one dimensional function space and interpretability is limited to the non-zero coefficients of the model.

While we are interested in determining a treatment recommendation, we are also interested in the question of characterizing the level sets of a treatment effect. Diffusion embeddings provide a non-linear framework to map out the data into a continuum of varying treatment effectiveness. Using a diffusion metric, one can determine variability of types of patients that similarly benefit from treatment (or lack of treatment).

Function regularized diffusion has been considered when F can be evaluated pointwise [13]. Intrinsic dimensionality has also been considered in the specific case that F is a monotonically increasing sampling of time [12, 14]. We have also previously examined building non-linear features of functions F that cannot be evaluated pointwise and subsequently organize these features [2], as well as regression of non-linear Cox proportional hazard functions [8].

The main contributions of this work are:

- the ability to build a function regularized diffusion metric without the ability to evaluate F pointwise,
- interpolation of a function regularized diffusion metric to new points where F is unknown,
- the use of a diffusion metric to define a filtration on a function F that is only evaluated on large subsets, and
- the application of localized filtrations to calculating a personalized treatment effectiveness ratio.

This paper is organized as follows. Section 2 gives background descriptions of the tools we reference throughout the paper, including diffusion maps, hierarchical cluster trees, and bigeometric organization. Section 3 details the function weighted trees used to generate ρ , as well as the introduces the notion of estimating a data point’s personalized function estimate. Section 4 applies and validates our algorithm on several datasets of synthetic patients, and discovers the original ground truth metric.

2. BACKGROUND

In this section, we discuss previous research that considers organization of points. This considers $M \in \mathbb{R}^{n \times m}$ as a data matrix of n points and m features per point. Denote the rows of M by X (the set of points), and the set of columns by Y (the set of features or questions). For this section, there is no external function f being considered.

2.1. Diffusion Geometry. Diffusion maps is a manifold learning technique based on solving the heat equation on a data graph [3]. It has been used successfully in a number of signal processing, machine learning, and data organization applications (CITE). We will briefly review the diffusion maps construction.

Let $X = \{x_1, \dots, x_n\}$ be a high dimensional dataset with $x_i \in \mathbb{R}^m$. A data graph is constructed with each point x_i as a node and edges between two nodes with weights $k(x_i, x_j)$. The affinity matrix $K_{i,j} = k(x_i, x_j)$ is required to be symmetric and non-negative. Common choices of kernel are the gaussian

$$k(x_i, x_j) = e^{-\frac{\|x_i - x_j\|_2^2}{2\sigma^2}},$$

or positive correlation

$$k(x_i, x_j) = \max \left(\frac{\langle x_i, x_j \rangle}{\|x_i\| \|x_j\|}, 0 \right).$$

K can be computed using only nearest neighbors of x_i such that $k(x_i, x_j) \geq \tau > 0$.

Let $D_{i,i} = \sum_j k(x_i, x_j)$. We normalize kernel K to create a Markov chain probability transition matrix

$$P = D^{-1}K.$$

The eigendecomposition of P yields a sequence of eigenpairs $\{(\lambda_i, \phi_i)\}_{i=0}^{n-1}$ such that $1 = \lambda_0 \geq \lambda_1 \geq \dots$

The diffusion distance $d_{DM}^t(x_i, x_j)$ measures the distance between two points as the probability of points transitioning to a common neighborhood in some time t . This gives

$$d_{DM}^t(x_i, x_j) = \sum_{x_k \in X} (P^t(x_i, x_k) - P^t(x_j, x_k))^2 = \sum_{k \geq 1} \lambda_k^{2t} (\phi_k(x_i) - \phi_k(x_j))^2.$$

Retaining only the first d eigenvectors creates an embedding $\Phi_t : X \rightarrow \mathbb{R}^d$ such that

$$\Phi_t : x_i \rightarrow [\lambda_1^t \phi_1(x_i), \dots, \lambda_d^t \phi_d(x_i)].$$

Figure 1 shows a two dimensional example dataset and the data graph generated on the points. We also see the low frequency eigenfunctions on the data graph, and the diffusion embedding Φ_t .

Remark: The diffusion time t is a continuous variable, which can be thought of as the degree to which Φ_t is a low-pass filter. For small t , more of the high-frequency eigenfunctions are given non-trivial weight. For large t , the embedding is mostly concentrated on the low-frequency eigenfunctions that vary slowly across the data.

2.2. Hierarchical Tree From Diffusion Distance. The main idea behind bigeometric organization is to construct a coupled geometry via a partition tree on both the data points and the features. A partition tree is effectively a set of increasingly refined partitions, in which finer child partitions (i.e. lower levels of the tree) are splits of the parent folder which attempt to minimize the inter-folder variability.

Let $X \subset \mathbb{R}^m$ be a dataset of points, and $\Phi_t : X \rightarrow \mathbb{R}^d$ be a diffusion embedding with corresponding diffusion distance $d_{DM}^t : X \times X \rightarrow \mathbb{R}^+$. A partition tree on X is a sequence of L tree levels \mathcal{X}^ℓ , $1 \leq \ell \leq L$. Each level ℓ consists of $n(\ell)$ disjoint sets \mathcal{X}_i^ℓ such that

$$X = \bigcup_{i=1}^{n(\ell)} \mathcal{X}_i^\ell.$$

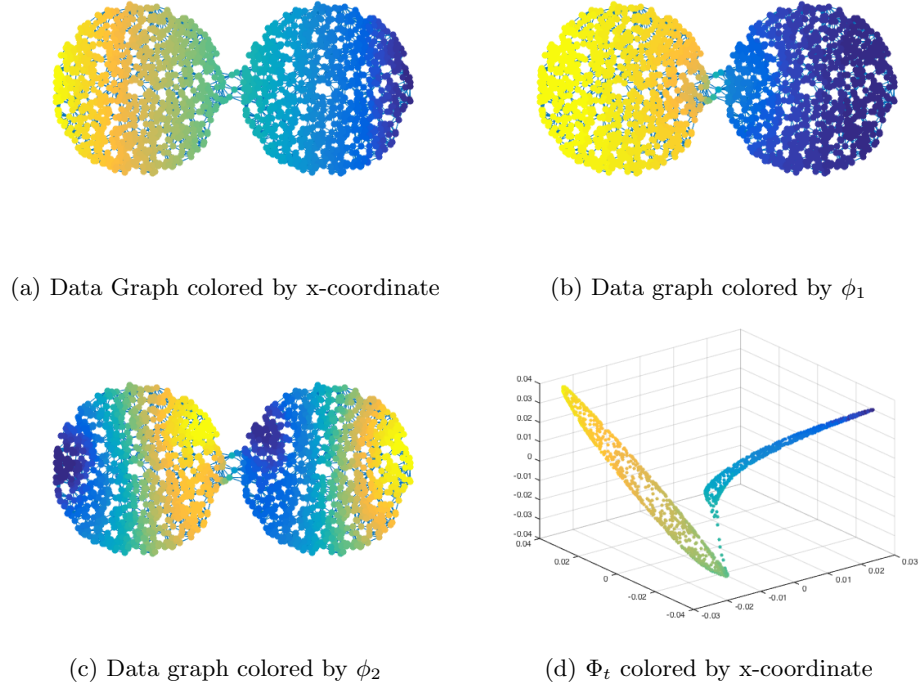


FIGURE 1. Toy example to demonstrate relationship between geometry of dataset and eigenfunctions of graph. Dataset is only 2D for ease of visualization, algorithm is equally valid on high-dimensional dataset.

Also, we define subfolders (or children) of a set \mathcal{X}_i^ℓ to be the indices $I_i^{\ell+1} \subset \{1, \dots, n(\ell+1)\}$ such that

$$\mathcal{X}_i^\ell = \bigcup_{k \in I_i^{\ell+1}} \mathcal{X}_k^{\ell+1}.$$

For notation, $\mathcal{X}^1 = X$ and $\mathcal{X}_i^L = \{x_i\}$. See Figure 2 for a visual breakdown of X .

This tree can be in two ways:

- (1) *Top-down:* Taking the embedded points $\Phi_t(X) \subset \mathbb{R}^d$, the initial split \mathcal{X}^2 divides the data into 2 (or k) clusters via k-means or some clustering algorithm. Each subsequent folder is then split into 2 (or k) clusters in a similar way, until each folder contains a singleton point.
- (2) *Bottom-up:* Taking the embedded points $\Phi_t(X) \subset \mathbb{R}^d$, the bottom folders \mathcal{X}^{L-1} are determined by choosing a fixed radius ϵ and covering $\Phi_t(X)$ with balls of radius

- ε. Each subsequent level of the tree is then generated as combinations of the children nodes that are “closest” together under the distance d_{DM}^t .

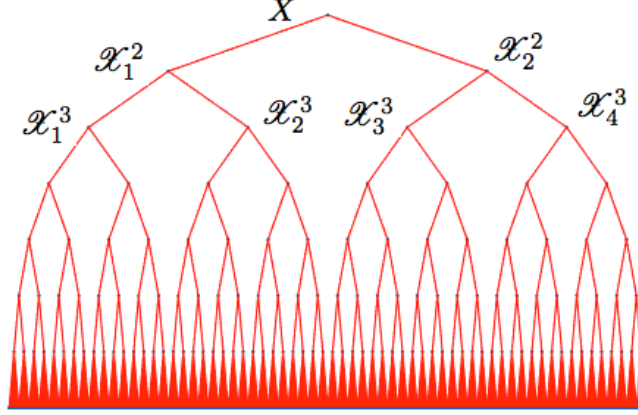


FIGURE 2. Breakdown of X into folders.

Remark: It is important to note that whether one chooses a top-down or bottom-up approach, the fact that the clustering occurs on the diffusion embedding $\Phi_t(X)$ makes the resulting tree, by definition, a “bottom-up” geometry. This is because the embedding and diffusion distance is built off of local similarities alone, meaning that the resultant geometry and partition tree are based on properties of the underlying dataset and manifold rather than the ambient dimension and a naive Euclidean distance in \mathbb{R}^m .

3. WEIGHTED AND DIRECTIONAL TREES WITHOUT POINTWISE FUNCTION EVALUATION

Let us denote our data space $X \subset \mathbb{R}^m$. In its most general form, we have an external function $F : X \rightarrow \mathbb{R}$ which cannot be evaluated pointwise. F can only be evaluated on large subsets $E \subset X$. We define the pointwise estimate of F to be $\hat{f}(x)$, as in (1). This is done by defining a type of locally weighted distance to incorporate estimates of a feature’s power to discriminate F in different half spaces. The details on this method are in Section 3.1. An overview of the approach is in Algorithm 1.

3.1. Weighted Trees. Let $M \in \mathbb{R}^{n \times m}$ be the data matrix and F be the integral operator of interest. Denote the rows of M by X (the set of points), and the set of columns by Y (the set of features or questions). We wish to build feature weights on each folder of \mathcal{X} that maximally separate F . The algorithm is as follows:

- (1) Assume the tree \mathcal{X} is known and separates X into hierarchical nodes. Fix a node \mathcal{X}_i^ℓ .

Algorithm 1 Calculate Function Weighted Metric

Required: Training points $\{x_i\}_{i=1}^n \in \mathbb{R}^m$

Function F to be evaluated on $E \subset \mathbb{R}^m$

Result: $\Phi_t : \{x_i\}_{i=1}^n \rightarrow \mathbb{R}^d$ such that $\text{range}(\hat{f}(x))$ is large, where

$$\begin{aligned} \hat{f} &: \{x_i\}_{i=1}^n \rightarrow \mathbb{R} \\ x &\mapsto F(\mathcal{N}_{d_{DM}^t}^\epsilon(x)) \end{aligned}$$

- (1) Build a diffusion embedding of the points $\Phi_t(X)$ and a hierarchical tree \mathcal{X}
 - (2) Build a tree \mathcal{Y} that determines the local coordinate feature weights (see Section 3.1)
 - (3) Build a new diffusion embedding of the points $\Phi_t(X)$ and a hierarchical tree \mathcal{X} based on the kernel in (5)
 - (4) Iterate between the points and the features until embedding $\Phi_t(X)$ and tree \mathcal{Y} are stable
 - (5) Define pointwise neighborhood $\mathcal{N}_{d_{DM}^t}^\epsilon(x) = \{z \in \mathbb{R}^m : \|\Phi_t(x) - \Phi_t(z)\|_2 < \epsilon\}$ and function estimate $\hat{f}(x)$
-

- (2) For each element $y \in Y$, we split y into k intervals $[a_j, a_{j+1})$ and bin the elements of $\mathcal{X}_i^\ell = \{h_j\}_{j=1}^k$ such that

$$x \in h_j \iff x(y) \in [a_j, a_{j+1}) \text{ and } x \in \mathcal{X}_i^\ell.$$

Question y is then assigned a local weight for its ability to discriminate F by

$$w_i^\ell(y) = \sum_{j=1}^k \frac{|h_j|}{|\mathcal{X}_i^\ell|} \cdot |F(h_j) - \bar{F}|^2, \quad (4)$$

where \bar{F} is the weighted mean across all bins.

- (3) Now that every node of the tree \mathcal{X} has local feature weights, we calculate the local weights at a point x_i by

$$w_{x_i}(y) = \sum_{\ell} 2^{-\alpha_\ell} w_{x_i}^\ell(y), \quad \text{where } w_{x_i}^\ell(y) = w_i^\ell(y) \text{ for } x_i \in \mathcal{X}_i^\ell.$$

These weights create a diagonal matrix W_{x_i} where $W_{x_i}[y, y] = (w_{x_i}(y) + \lambda)^{-1}$ for a small positive constant λ .

- (4) The kernel function $k : X \times X \rightarrow \mathbb{R}^+$ is then

$$k(x_i, x_j) = \frac{e^{-(x_i - x_j)^\top (W_{x_i} + W_{x_j})^{-1} (x_i - x_j) / \sigma^2}}{\sqrt{\det(W_{x_i} + W_{x_j})}}. \quad (5)$$

The normalization in the denominator is needed to guarantee k is positive semi-definite.

Theorem 3.1. $k : X \times X \rightarrow \mathbb{R}^+$ from (5) is positive semi-definite.

Proof. We show k can be expressed as an integral over all ambient space via

$$k(x, y) = \int dz \frac{\exp\{-(x-z)^\top W_x^{-1}(x-z)/\sigma^2\}}{\sqrt{\det(W_x)}} \frac{\exp\{-(y-z)^\top W_y^{-1}(y-z)/\sigma^2\}}{\sqrt{\det(W_y)}}.$$

We then use an identity from [10] which the product of two gaussians gives

$$\frac{C}{\sqrt{\det(W_x + W_y)}} \exp\left[-(x-y)^\top (W_x + W_y)^{-1}(x-y)/\sigma^2\right] \cdot \frac{e^{-m^\top W^{-1}m}}{\sqrt{\det(W)}}.$$

where m and W are combinations of x , y , and z . Their exact forms are irrelevant, as the right hand term is simply a normalized gaussian that can be integrated out with respect to z . Thus, after evaluating the integral, we are left with

$$k(x, y) = C \cdot \frac{e^{-(x-y)^\top (W_x + W_y)^{-1}(x-y)/\sigma^2}}{\sqrt{\det(W_x + W_y)}}.$$

Now, for any $w(x)$ we can compute

$$\begin{aligned} \int dx dy w(x)w(y)k(x, y) &= \int dx dy w(x)w(y) \int dz \frac{\exp\{-(x-z)^\top W_x^{-1}(x-z)/\sigma^2\}}{\sqrt{\det(W_x)}} \frac{\exp\{-(y-z)^\top W_y^{-1}(y-z)/\sigma^2\}}{\sqrt{\det(W_y)}} \\ &= \int dz \left(\int dx w(x) \frac{\exp\{-(x-z)^\top W_x^{-1}(x-z)/\sigma^2\}}{\sqrt{\det(W_x)}} \right) \left(\int dy w(y) \frac{\exp\{-(y-z)^\top W_y^{-1}(y-z)/\sigma^2\}}{\sqrt{\det(W_y)}} \right) \\ &= \int dz \left(\int dx w(x) \frac{\exp\{-(x-z)^\top W_x^{-1}(x-z)/\sigma^2\}}{\sqrt{\det(W_x)}} \right)^2 \\ &\geq 0. \end{aligned}$$

□

Because k is semi-positive definite, we can compute the embedding of the data $\Phi_t(X)$, and induce a new diffusion metric on the data,

$$d_{DM}^t(x, y) = \|\Phi_t(x) - \Phi_t(y)\|_2.$$

This, in turn, allows us to define $\hat{f} : \{x_i\} \rightarrow \mathbb{R}$ as an estimate to $f(x_i)$, where

$$\hat{f}(x) = F(\mathcal{N}_{d_{DM}^t}^\epsilon(x)), \quad \text{where } \mathcal{N}_{d_{DM}^t}^\epsilon(x) = \{z \in \{x_i\} : \|\Phi_t(x) - \Phi_t(z)\|_2 < \epsilon\}.$$

3.2. Interpolation and Leave Out Validation. The metric d_{DM}^t and function estimate \hat{f} can easily be extended to new points $z \notin \{x_i\}$ not in the training data. This is done by building an asymmetric affinity matrix to the training data, which can be thought of as a reference set. The approach is an application of [9], which we briefly outline here.

Let X be training data, and F defined on subsets of X . Let Z be testing points on which F is not defined a priori. Define $k : (X \cup Z) \times X \rightarrow \mathbb{R}^+$ to be

$$k(z, x) = \frac{e^{-(z-x)^\top W_x^{-1}(z-x)/\sigma^2}}{\sqrt{\det(W_x)}}, \quad z \in Z \cup X, x \in X.$$

With the normalization matrices $(D_1)_{ii} = \sum_j k(z_i, x_j)$ and $(D_2)_{ii} = \sum_j k(z_j, x_i)$, we set

$$A = D_1^{-1/2} k D_2^{-1/2},$$

and take the eigendecomposition of the small matrix $A^* A = \Psi \Sigma \Psi^*$. This gives an embedding of the reference points X . Then eigendecomposition of the entire set of points $X \cup Z$ is estimated by

$$\Phi = A \Psi.$$

The details of the extension algorithm can be found in Algorithm 2.

Algorithm 2 Nearest Neighbor Function Estimation

Required: Training points $\{x_i\}_{i=1}^n \in \mathbb{R}^m$
Function F to be evaluated on $E \subset \{x_i\}$
Testing points $\{z_i\}_{i=1}^N \in \mathbb{R}^m$
Result: Pointwise function estimate on testing points $\hat{f}(z_i)$

- (1) Build stable function weighted diffusion embedding of the features $\Phi_t(Y)$ and a hierarchical tree \mathcal{Y} via Algorithm 1 using the training points $\{x_i\}_{i=1}^n$
- (2) Build a function weighted diffusion embedding $\Phi_t(X \cup Z)$ using the weighted embedding via reference set algorithm (see Section 3.2)
- (3) For each testing point z , define the pointwise training neighborhood $\mathcal{N}_{d_{DM}^t}^\epsilon(z) = \{x \in \{x_i\} : \|\Phi_t(x) - \Phi_t(z)\|_2 < \epsilon\}$ and function estimate

$$\hat{f}(z) = F(\mathcal{N}_{d_{DM}^t}^\epsilon(z))$$

Algorithm 2 can be thought of as generating an optimized metric for a k-nearest neighbor search. There could be better mechanisms of classification and regression for predicting \hat{f} , ranging from support vector machines [1] to various types of linear regression [15]. These choices are application and function specific, which is why we remain with a simple nearest neighbor interpolation. The key is that the metric d_{DM}^t agrees with the intrinsic geometry of the data.

It is also important to run leave out validation of the algorithm to insure no overfitting of the data. As the algorithm is semi-supervised and weights variables according to their discriminatory power, it is possible to give high weight to features which are spuriously correlated with the function. This makes it crucial to run N-fold cross validation of the data to ensure that the predicted $\hat{f}(z)$ are good estimates to the true function.

4. APPLICATIONS

4.1. Semi-supervised Bernoulli Random Variable. We create a model of observational data for synthetic patients. The patient baseline model consists of 9 dimensions of uncorrelated information, with $x_k \in [0, 1]$. The hazard function for the patients is a bernoulli random variable Z_X , where

$$P(Z_X = 1|X) = G(x_1, x_2),$$

with G being the gaussian displayed in Figure 3.

Furthermore, we assume that many of the patients have an unknown outcome. We model this by assigning a random variable ξ_X for each patient such that $P(\xi_X = 1|X) = P(\xi_X = 0|X) = 0.5$. We only observe the function

$$f(x) = \begin{cases} Z_X, & \xi = 1 \\ NaN, & \xi = 0 \end{cases},$$

where NaN implies the outcome is unknown. After dropping 50% of the known function values for our $N = 10,000$ patients, we are left with only observing 392 outcomes in which $Z_X = 1$.

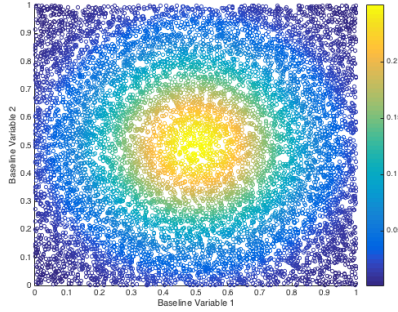


FIGURE 3.

A logistic regression is unable to recover the nonlinear G . But beyond that, in this case a linear model fails to even recover the significant factors to any degree of accuracy. See Table 1 for the regression coefficients. Note that the linear model can only be run on patients for which $\xi_X = 1$.

Variable Name	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
Coefficient	-0.0027	-0.0159	0.0369	-0.0144	0.0067	-0.0044	-0.0092	-0.0286	0.0307
p-value	0.9436	0.6769	0.3327	0.7069	0.8609	0.9092	0.8087	0.4529	0.4233

TABLE 1.

Our algorithm on the other hand is able to recover both the geometry of the patients and an estimate of the personalized risk $G(x)$. We do this by evaluating the function

$$F(E) = \mathbb{E}_{\{X:\xi_X=1\}}[f(X)]$$

on large bins of data, specifically splitting each feature into three bins. Figure 4 shows the recovered embedding of the patients, and is colored by the estimate of $F(x)$.

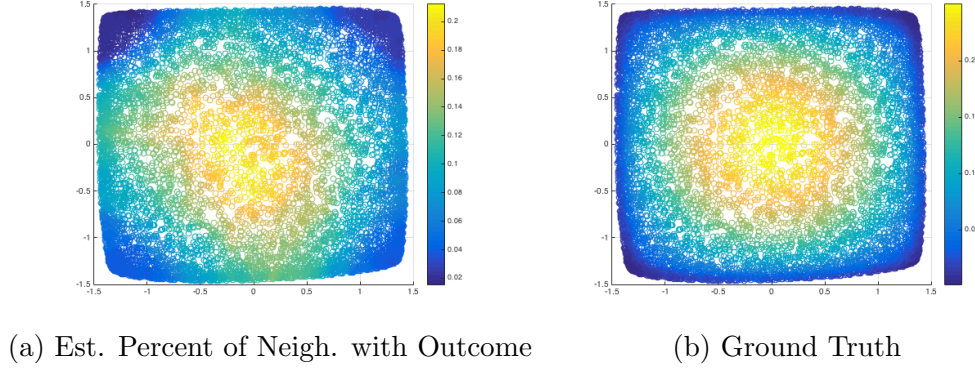


FIGURE 4.

The model works in a predictive fashion, as well. Because the patients with $\xi_X = 0$ have unobserved outcomes, their placement in the embedding is unbiased by the value of $f(X)$. We bin these 5000 patients into groups of level sets of $G(X)$, and look at the average predicted risk in that neighborhood. The results are shown in Figure 5.

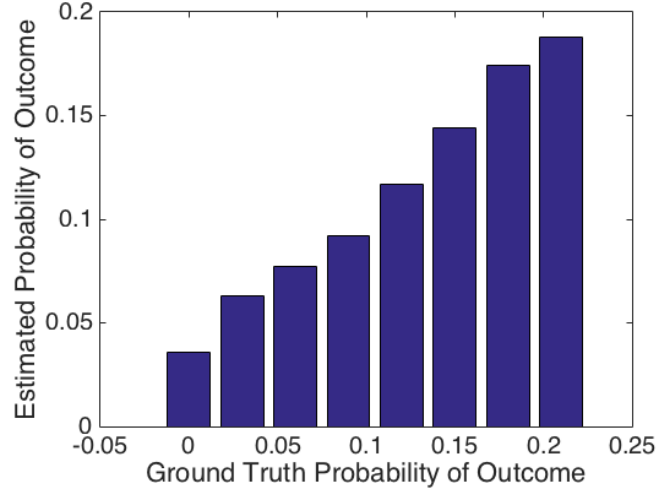


FIGURE 5.

4.2. Synthetic Randomized Drug Trial. We create a model of synthetic patients in a drug trial. The patient baseline model consists of 9 dimensions of correlated information, where

$$x_k \in [0, 1], \quad x_{3i+1}^2 + x_{3i+2}^2 + x_{3i+3}^2 = 1, \quad \text{for } i \in \{0, 1, 2\}.$$

We note that this model choice is arbitrary, and was solely chosen to model a dependence between patient features. The patients are randomly split into treatment A and treatment B.

The baseline hazard function for the patients is a Weibull distribution of the type

$$P(X < t) = e^{-\lambda t^k},$$

for $\lambda = 2$ and $k = 1.2$. If a patient is in treatment A, their outcome time t_x is sampled from the Weibull distribution. If a patient is in treatment B, their outcome time t_x is sampled from the Weibull distribution and then adjusted by $t_x \mapsto t_x e^{\beta_x}$. Any patient is censored if $t_x > T$ for a fixed T .

The key behind this model is that β_x is patient specific, and depends only on a subset of the patient's baseline information. Specifically,

$$\beta_x = f(x_1, x_2, x_3),$$

where f is displayed in Figure 6. Setting $T = 2$ and $\sup |e^{\beta_x}| = 3$, about 10.5% of patients have an outcome.

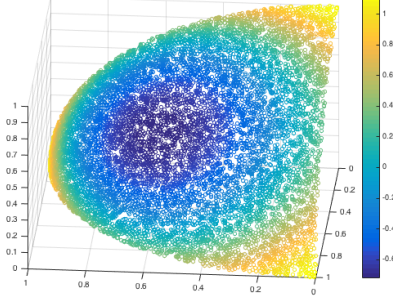


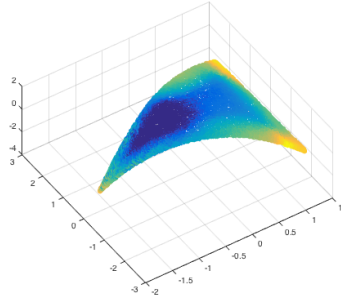
FIGURE 6.

A linear Cox proportional hazard model, by definition, is unable to recover the full spread of β_x . But beyond that, in this case a linear model fails to even recover the treatment group as a significant factor in risk. See Table 2 for the regression coefficients.

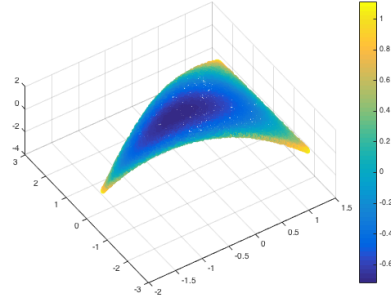
Our algorithm on the other hand is able to recover both the geometry of the patients and an estimate of the personalized hazard ratio β_x . Figure 7 shows the recovered embedding of the patients, and is colored by the estimate of β_x . The model works for predictive personalized hazards on new testing data, as well. We run repeated random sub-sampling

Variable Name	Treatment	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
Coefficient	0.0597	-1.8517	-1.6500	-1.8440	-0.1797	-0.3245	-0.1619	-0.1127	0.0566	0.0954
p-value	0.3155	0.0001	0.0001	0.0001	0.3766	0.1087	0.4295	0.5826	0.7854	0.6450

TABLE 2.



(a) Predicted HR



(b) Ground Truth HR

FIGURE 7.

validation on the toy data by retaining 80% of the patients for training, and testing on the remaining 20%. This was iterated 100 times. The results are shown in Figure 8.

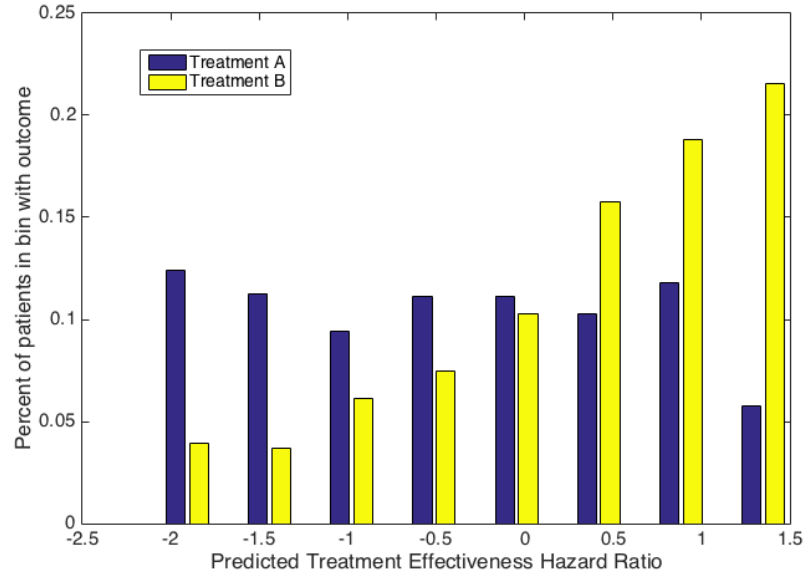


FIGURE 8.

4.3. Models with Treatment Propensity. Let $X \sim \mathcal{N}(0, \Sigma)$, with

$$\Sigma_{i,j} = \begin{cases} 1, & \text{if } i = j \\ 0.5, & \text{if } |i - j| = 1 \\ 0, & \text{otherwise} \end{cases}.$$

The baseline hazard function is the same as in Sections 4.1 and 4.2, and the personal hazard ratio is

$$h(X) = X_1 + 0.5X_2 + 0.5X_1X_2 + X_\alpha X_2.$$

However, unlike in the previous examples, X_α is not randomized across the population. Instead,

$$P(X_\alpha = 1) = P(w_x < \gamma_0 + X\gamma),$$

where $w_x \sim \mathcal{N}(0, 1)$, $\gamma_0 = 0.5$, and $\gamma = [1 \ 1 \ 0 \ \dots \ 0]$.

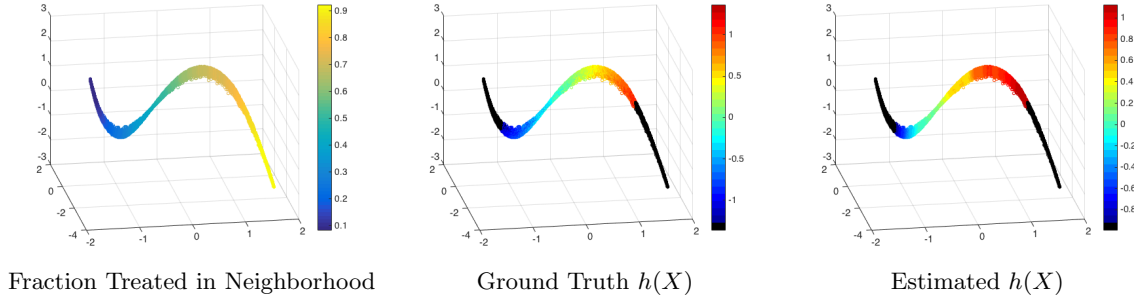


FIGURE 9. Black corresponds to points where $> 80\%$ of the points were from the same treatment group, and thus removed due to lack of estimate precision.

We also consider a random model, in which Σ is a random symmetric positive definite matrix with condition number less than 10, and the personal hazard ratio follows the form

$$h(X) = \sum_i \xi_i X_i + \sum_{i,j} \eta_{i,j} X_i X_j + X_\alpha \left(\sum_i \nu_i X_i + \sum_{i,j} \delta_{i,j} X_i X_j \right),$$

where ξ_i and ν_i are sparse standard normal random variables which are non-zero with probability 0.5. Also, $\eta_{i,j}$ (resp. $\delta_{i,j}$) are standard normal random variables which are non-zero if and only if ξ_i and ξ_j (resp. ν_i and ν_j) are non-zero. Also, the probability that a patient is treated is determined by

$$P(X_\alpha = 1) = P(w_x < \gamma_0 + X\gamma).$$

We run this model across 100 iterations, where we generate 2,000 patients whose baseline hazard function is drawn from a Weibull distribution as in previous examples. The

patients are then censored such that ϵ fraction of the patients have an outcome, where ϵ is a uniform random variable drawn from $[1/3, 1]$. We calculate the correlation between the predicted personalized treatment effect $\hat{f}(x)$ and the ground truth treatment effect $\sum_i \nu_i X_i + \sum_{i,j} \delta_{i,j} X_i X_j$. Because of the propensity for treatment, we only estimate $\hat{f}(x)$ in neighborhoods such that $\leq 80\%$ of the patients are in the same treatment group. The histogram of correlations is in Figure 10.

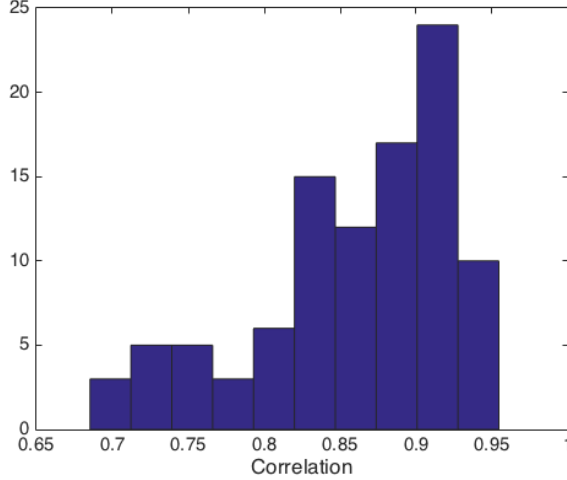


FIGURE 10. Histogram of Correlations between predicted personalized treatment effect and ground truth across 100 iterations.

5. CONCLUSIONS AND FUTURE WORK

This paper develops a method for building a data dependent metric $\rho : X \times X \rightarrow \mathbb{R}^+$ that is simultaneously learns the level sets of a function F . The method only needs to evaluate F on various half spaces of the data, making it useful when F cannot be evaluated pointwise. We develop a weighted tree distance to accomplish this, and simultaneously organize both the points and the features. Once ρ has been discovered, we can do k nearest neighbor prediction for new points added to the data without knowledge of F at the point.

This algorithm was designed with medical applications in mind, specifically building a local cox proportional hazard model for patients in a dataset. The embedding created by ρ can be used to characterize types of people that are hurt or helped by a drug, and even assign a personalized treatment hazard score to new patients whose outcomes are unknown.

ACKNOWLEDGEMENTS

The author would like to thank Raphy Coifman for many discussions about the problem, Harlan Krumholz, Shu-Xia Li, and Nihar Desai for introducing the issues associated with treatment effectiveness and Cox proportional hazard models, and Fred Warner, Andreas Coppi, and Chenxi Huang for discussions about code and algorithm implementation. The author is supported by NSF Award No. DMS-1402254.

REFERENCES

- [1] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT '92*, pages 144–152, New York, NY, USA, 1992. ACM.
- [2] Alexander Cloninger, Ronald R Coifman, Nicholas Downing, and Harlan M Krumholz. Bigeometric organization of deep nets. *Applied and Computational Harmonic Analysis*, 2016.
- [3] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.
- [4] D.R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B*, 1972.
- [5] Holly Janes, Marshall D Brown, Ying Huang, and Margaret S Pepe. An approach to evaluating and comparing biomarkers for patient treatment selection. *The international journal of biostatistics*, 10(1):99–121, 2014.
- [6] Holly Janes, Margaret S Pepe, Patrick M Bossuyt, and William E Barlow. Measuring the performance of markers for guiding treatment decisions. *Annals of internal medicine*, 154(4):253–259, 2011.
- [7] Chaeryon Kang, Holly Janes, and Ying Huang. Combining biomarkers to optimize patient treatment recommendations. *Biometrics*, 70(3):695–707, 2014.
- [8] Jared Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. Deep survival: A deep cox proportional hazards network. *arxiv.org/abs/1606.00931*, 2016.
- [9] Dan Kushnir, Ali Haddad, and Ronald R Coifman. Anisotropic diffusion on sub-manifolds with application to earth structure classification. *Applied and Computational Harmonic Analysis*, 32(2):280–294, 2012.
- [10] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7:15, 2008.
- [11] Min Qian and Susan A Murphy. Performance guarantees for individualized treatment rules. *Annals of statistics*, 39(2):1180, 2011.
- [12] Amit Singer and Ronald R Coifman. Non-linear independent component analysis with diffusion maps. *Applied and Computational Harmonic Analysis*, 25(2):226–239, 2008.
- [13] Arthur D Szlam, Mauro Maggioni, and Ronald R Coifman. Regularization on graphs with function-adapted diffusion processes. *The Journal of Machine Learning Research*, 9:1711–1739, 2008.
- [14] Ronen Talmon and Ronald R. Coifman. Empirical intrinsic geometry for nonlinear modeling and time series filtering. *Proceedings of the National Academy of Sciences*, 110(31):12535–12540, 2013.
- [15] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *J. R. Statist. Soc. B*, 2005.